2 Overall Process

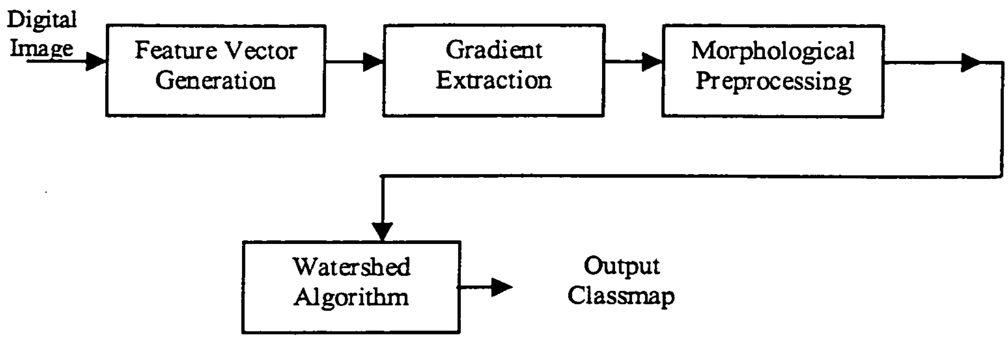

Figure 1 Overall process used in region extraction

3 Input Image Data

The digital input images are assumed to be in YUV format. If the inputs are in a chrominance sub-sampled format such as 420, 411 or 422, the chrominance data is upsampled to generate 444 material.

4 Feature Vector Generation

We extract one feature vector for each PxQ block of the input picture. There are two stages in the feature vector generation process. In the first stage, we transform the data from the original YUV color co-ordinate system into another co-ordinate system known as $CIE-L^*a^*b^*$ [see *Fundamentals of Digital Image Processing*, by Anil K. Jain, Prentice-Hall, Section 3.9]. The latter is known to be a perceptually uniform color system, i.e. the Euclidean distance between two points (or colors) in the $CIE-L^*a^*b^*$ co-ordinate system corresponds to the perceptual difference between the colors.

The next stage in the feature vector generation process is the calculation of the first N moments of the $CIE-L^*a^*b^*$ data in each block. Thus, each feature vector has 3N components (N moments in L, N moments in a, and N moments in b). We can denote the (3Nx1) feature vector of the (i, j)-th block of the input picture as follows.

$$\vec{f}(i,j) = [{}_L m_1, \ldots, {}_L m_N, {}_a m_1, \ldots, {}_a m_N, {}_b m_1, \ldots, {}_b m_N]^T,$$

where the k-th moment in, say, the L component, is given by

$${}_L m_k = \frac{1}{PQ} \sum L^k(x,y),$$

where (x, y) represents the index of a point in the (i, j)-th block.

5 Gradient Extraction

The next stage in our region extraction process is that of gradient extraction. We will estimate a block-based gradient field for the input picture (i.e. we get one scalar gradient value for each PxQ block of the input picture). The gradient at the (i, j)-th block of the input picture is defined as the maximum of the distances between the block's feature vector $\vec{f}(i,j)$ and its nearest neighbor's feature vectors.

$$grad(i,j) = \max_{k,l \in \{-1,0,1\}} \{d[\vec{f}(i,j), \vec{f}(i-k, j-l)]\},$$

where d[.,.] is function that assigns a distance value to a pair of feature vectors. (Note: in the above maximization, we let k and l each vary from -1 to $+1$, but do not allow $k = l = 0$ simultaneously! Also, along the borders of the image, we consider only those neighboring blocks that lie inside the image boundaries). In our work, we will employ two types of distance functions.

We could use other methods to select the gradient value from the above set of distances, for example the minimum, median, etc. We need to evaluate the performance of the segmentation algorithm when such methods are used.

5.1 Weighted Euclidean Distance Metric

Here, the distance function d[.,.] is simply the weighted Euclidean distance between the two vectors.

$$d[\vec{f}(i,j), \vec{g}(k,l)] = \sqrt{\begin{array}{l} \{{}_L w_1 ({}_L m_{1,f} - {}_L m_{1,g})^2 + \ldots + {}_L w_N ({}_L m_{N,f} - {}_L m_{N,g})^2 + \\ {}_a w_1 ({}_a m_{1,f} - {}_a m_{1,g})^2 + \ldots + {}_a w_N ({}_a m_{N,f} - {}_a m_{N,g})^2 + \\ {}_b w_1 ({}_b m_{1,f} - {}_b m_{1,g})^2 + \ldots + {}_b w_N ({}_b m_{N,f} - {}_b m_{N,g})^2\} \end{array}}, \text{ where}$$

$$\vec{g}(k,l) \equiv \vec{f}(i = k, j - l).$$

In the above formula, the weighting factors $\{{}_{\{\}} w_{\{\}}\}$ can be used to account for the differences in scale among the various moments. This metric is very easy to implement. In our implementation, we set N = 1, i.e. use only the mean values within each PxQ block, and set the weighting factors to unity (this makes sense, since the $CIE - L^* a^* b^*$ space is perceptually uniform).

5.2 Probability Mass Function Based Distance Metric

The second choice of the distance metric is a little more involved. Here, we exploit the fact that using the moments of the data within the PxQ block, we can compute an approximation to the probability mass function (pmf) of that data. The pmf essentially describes the distribution of the data to be composed of a mixture of several values $v_0, v_1, v_2, \ldots$, with respective probabilities $P_0, P_1, P_2, \ldots$ The values and the probabilities together constitute the pmf. We can compute these values using the moments as follows. For ease of notation, we will drop the subscripts *L, a,* and *b,* because the equations that we provide apply to all three color components.

Initially, we approximate the distribution as a mixture of two values, v_0 and v_1, with probabilities P_0 and P_1 respectively. We use the moments-based approach given in Ali Tabatabai's Ph.D. thesis to estimate the values v_0, v_1, P_0 and P_1. In this method, we need the first three moments of the data (i.e. N = 3):

$$m_1 = \frac{1}{PQ}\sum L(x,y),$$

$$m_2 = \frac{1}{PQ}\sum L^2(x,y), \text{ and}$$

$$m_3 = \frac{1}{PQ}\sum L^3(x,y),$$

where $L(x,y)$ are data values in the (i,j)-th block. Then,

$$P_0 = \frac{1}{2}\left(1 + s\sqrt{\frac{1}{4+s^2}}\right),$$

$$P_1 = 1 - P_0,$$

$$v_0 = m_1 - \sigma\sqrt{P_1/P_0}, \text{ and}$$

$$v_1 = m_1 + \sigma\sqrt{P_0/P_1}, \text{ where}$$

$$s = \frac{m_3 + 2m_1^3 - 3m_1m_2}{\sigma^3}, \text{ and}$$

$$\sigma = \sqrt{\{m_2 - (m_1)^2\}}.$$

Thus, we can convert the moment-based feature vector of each PxQ block into a pmf-based representation. Once we have such a representation, then the distance between two feature vectors can be computed via the distance between the two pmf's. For this, we make use of the Kolmogorov-Smirnoff (K-S) test, as described in Section 14.3 of *"Numerical Recipes in C"*, 2nd edition, by W. A. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Cambridge University Press. (Essentially, the distance between two pmf's is the area under the absolute value of the difference between the two cumulative distribution functions, see the above-mentioned chapter for details).

Though the K-S test is prescribed for pmf's of a single variable, the data we have is in fact three-dimensional (*L, a,* and *b* components). Strictly speaking, we need to compute the joint, three-dimensional pmf, and then compute a distance between two pmf's. This is however a very hard problem to solve, and instead, we make a simplifying assumption. We assume that the color data in a PxQ block can be modeled by means of three independent pmf's, one each for the *L, a,* and *b* components. Let us denote these pmf's by pmf_L, pmf_a, and pmf_b respectively. Also, denote the K-S distance measure between two pmfs by $d_{KS}(.,.)$, then, the overall distance metric is given by

$$d[\vec{f}(i,j), \vec{g}(k,l)] = d_{KS}(pmf_{L,f}, pmf_{L,g}) + d_{KS}(pmf_{a,f}, pmf_{a,g}) + d_{KS}(pmf_{b,f}, pmf_{b,g}).$$